社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

Symmetry Reductionを使ったAISの確率付きモデル検査

豊島　崇士†　　高橋　和子†　　関澤　俊弦††

† 関西学院大学
†† 大阪学院大学/産業技術総合研究所 システム検証研究センター

あらまし　船舶自動識別システム (AIS) の予約システムについて，確率付きモデル検査器 PRISM を使った解析と検証を行う．このシステムでは予約方法に確率的要素が含まれており，モデル検査時に確率計算を行うため，モデルの規模が大きくなって状態数が増えるとすぐに計算量的爆発を起こす．本研究では，このモデルの対称性をいかしたSymmetry Reduction を適用することで状態数を削減し，規模の大きなモデルでの検証に成功した．また，同じ仕様を満たす確率でも，モデルの規模が大きくなると，初期状態に依存して大きく変動することが判明した．
キーワード　確率付きモデル検査, PRISM, Symmetry Reduction, AIS

Probabilistic Model Checking for AIS with Symmetry Reduction

Takashi TOYOSHIMA†, Kazuko TAKAHASHI†, and Toshifusa SEKIZAWA††

† Kwansei Gakuin University
†† Osaka Gakuin University/National Institute of Advanced Industrial Science and Technology，Research Center for Verification and Semantics

Abstract　We show the analysis and verification for the reservation system used in an Automatic Identification System (AIS) with a probabilistic model checking tool PRISM. This system includes a probabilistic factor, and probability is calculated on model checking. It follows that model checking soon falls into the computational explosion when the state increases in number according to the model size. We have succeeded in verification for larger case studies by applying symmetry reduction technique which takes an advantage of the symmetry of the model for AIS. Moreover, we have found that the probability for the same specification awfully changes depending on the initial state when a model size is large.
Key words　Probabilistic Model Checking, PRISM, Symmetry Reduction, AIS

1. まえがき

2000 年，国際海事機関（IMO：International Maritime Organization）の海上安全委員会により，平成 14 年 (2002 年)7 月から平成 20 年 (2008 年)7 月までに，データ通信機能を具備した AIS の旅客船舶や外航の貨物船舶，漁船舶などへの順次搭載が義務化された．AIS とは船舶に搭載することで識別符号，船舶名，位置，針路，船舶の速度といったその船舶の航海に関する情報を自動的かつ周期的に他の船舶や陸上の施設へ送信，他の船舶からの情報の受信をする装置のことである [1].

AIS の通信方式では同時に 2 隻以上の船舶が情報を送信しようとすると，その情報が他の船舶に届かない状況になる．そこで，AIS では情報を送信すると同時に，次に送信する時間の予約を行う．予約を行うことで同時に多数の船舶が情報の送信を行うことを回避する．

しかし，この予約システムに対して検証を行われた例がなく，多数の船舶が同じ時間に対して送信の予約をしてしまった結果，同時に情報の送信を行う場合がないとは言い切れない．

実際の AIS では予約の重なりが起こったとしても，同じ時間間隔で予約を続けるわけではなく，しばらく時間が経過すると別の時間間隔をとって予約を行い，その時に予約の重なりが解消される．また，時間間隔のとり方には多くのバリエーションがあるため，2 隻以上の船舶の予約が連続して重なることは確率的にほぼない．したがって，複雑なプロトコルによって安全を保証するよりも，一度くらいの予約の重なりは無視し，確率的に安全であれば容認されている．我々は，確率付きモデル検査器 PRISM [2] を用いて「予約が重なることなく取れるか」が確率的に安全であるかについて解析と検証を行い，結果として許容範囲であるということを示した [3].

しかし，作成したモデルは船舶の数を増やすとすぐに状態爆発を起こしてしまうため，船舶の数が 3 というとかなり規模の小さい場合の検証しか行えなかった．

そこで，本研究では Probabilistic Symmetry Reduction を施すことで状態数を削減したモデルを生成し，モデルの規模を大きくした場合の検証を行う．

Symmetry Reduction は，同値関係にある状態をまとめて 1 つの状態とみなすことで状態数削減を行う手法であり，Petri-net や，LTS(Labeled Transition Sysmtem) などの状態遷移系に適用されており [4] [5]，モデル検査器でも成功している例は多い [6] [7]．Probabilistic Symmetry Reduction は状態から状態へ遷移する際の確率も含めて同値類をとるものであり，物理現象を対象としたもの [8] や，ディスクシステムを対象としたもの [9] があるものの，その応用例は少ない．

本研究では AIS の予約システムの状態遷移系に対して「回転」と「反転」という操作を定義し，これらの操作によって一致させることのできる状態同士を同値とみなして Probabilistic Symmetry Reduction を行った．AIS モデルでは Reduction の前後で対応する状態遷移間の確率は等しいという特徴を持つ．

Reduction を行った結果，船舶の数を増やした検証に成功し，状態数も多いものでは 4 桁分抑えることができた．また，Reduction を施したモデルに対する検証の結果，船舶の数が増えれば予約の重なる確率は上昇してしまうことがわかった．この結果は予約の重なる確率は船舶の数に関らないという一般的な予想に反するものであり，船舶の数が少ない場合のみの検証では得られなかったものである．この結果から確率付きのシステムに対しては必ずしもモデルの規模によって仕様を満たす確率が等しいとは限らないことが判明した．

本論文の構成は以下の通りである．2 節で検査対象とする AIS における送信予約システムを説明し，3 章で確率付きモデル検査及び PRISM について記述する．4 章で送信予約システムのモデル化を説明し，5 章で本研究で行った Symmetry Reduction を記述し，6 章に Reduction を行ったモデルに対する検証と考察について示す．7 章にまとめと今後の課題について述べる．

2. AIS における送信予約システム

AIS は他の船舶や陸上の施設と情報を交換し合う通信機能を持つが，その送受信機はいわばトランシーバーのようなもので，自船舶が情報の送信を行っている時に他の船舶が送信した情報が受信できない状態となる．

AIS ではそういった状況を改善するために，SOTDMA(Self Organized Time Division Multiple Access) 方式という通信方式がとられている．この方式ではまず 1 分を 1 フレームと定義し，そのフレームを 2250 のタイムスロットに分割する．1 スロットの長さは 26.7ms となり，このスロットにあたる時間に情報を送信するのだが，その送信を行うのと同時に次にどのスロットで情報送信を行うかもその情報の中に含ませ，次に送信するスロットの予約を行う．

この方式では，各船舶が空きスロットを確認してから次フレームに予約を入れるまでにはタイムラグが発生する場合がある．そのため，複数の船舶が空きスロットと判断して同一のスロットに予約を入れてしまう場合がある．複数の船舶が予約していた時間に同時に情報の送信を行うことをバーストと言う．

バースト時にはいくつかの問題点がある．1 つは送信側は互いに相手の情報を受信できなくなる問題．2 つ目は受信側も送信されてくる情報を受信できなくなる問題で，この時，受信側はどの船舶の送信が受信できなかったのか全く分からないことも問題となる．3 つ目としては送信側は送信が重なっていることに気づけないため，他の船舶へ情報送信が失敗していることに気づけない問題がある．つまり，バースト状態になると予約スロットをいつまでも修正することができない可能性がある．

情報送受信が正しく行われずにそのままでいることはとても危険な状態であるため，バースト状態は回避しなければならない．

3. 確率付きモデル検査

モデル検査ではシステムの有限状態モデルとそのシステムが満たすべき仕様として記述された論理式を与えられた時，システムがその仕様を満たしているかどうかを網羅的に検査する．仕様は論理式を使って記述されることが多い [10]．

本研究では確率を含んだシステムの動作を扱うために確率付きモデル検査器を用いることにする．確率付きモデル検査器には PRISM [2]，APMC [11]，ETMCC [12]，YMER [13] などがあるが，本研究では GUI の使いやすさ，ユーザの多さの点から PRISM を用いる．

PRISM はバーミンガム大学と現オックスフォード大学が開発した確率付きモデル検査器である．PRISM を用いることで確率を含んだ動作を形式的にモデル化し，その動作の解析と検証を行うことが可能である．仕様は確率付き時相論理 PCTL(Probabilistic real time Computation Tree Logic) [14] で与えられる(注1)．

PCTL は CTL(Computation Tree Logic) [10] を拡張し，確率的な検証を行える時相論理の一つである．

PCTL の構文規則は以下のようになる．

$\phi ::= \alpha \mid \phi \mid \phi_1 \& \phi_2 \mid \phi_1 \ ' \mid ' \ \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \mathrm{P}_{\beta p}[\psi]$

$\psi ::= \mathrm{X}\phi \mid \phi_1 \mathrm{U}^{\leqq\gamma} \phi_2 \mid \mathrm{G}^{\leqq\gamma}\phi \mid \mathrm{F}^{\leqq\gamma}\phi$

ϕ は state formula，ψ は path formula を表している．α は原子命題，p は確率を表しており範囲は $p \in [0,1]$，β は関係演算子を表し $\beta \in \{\leqq, <, >, \geqq\}$ である．

X,U,G,F は時相演算子と呼ばれるもので，CTL と同様単一パスにおける検査を記述するのに使用する．

また，PCTL では有限回遷移した場合と，無限回遷移した場合いずれにおける確率も求めることができる．PRISM では状態から状態への遷移を 1 ステップとしており，PCTL 式において X,U,G,F のような時相演算子を使用する際には，γ に有限回のステップを表す特定の自然数を代入する．また，それぞれの演算子の後の $^{\leqq\gamma}$ を取りはずすと無限回遷移する場合が表現できる．

path formula の意味はそれぞれ以下の通りである．

- $\mathrm{X}\phi$: ある状態の次の状態が ϕ を満たす．X ではステッ

(注1)：PRISM で扱える仕様は PCTL の他にもモデルによって CSL や LTL により記述可能．

プ数が 1 と決まっているので γ を指定する必要はない．

- $\phi_1 \mathrm{U}^{\leqq\gamma} \phi_2$ ： ある状態から γ ステップ以内の状態において ϕ_2 が満たされていると，それ以前の状態で ϕ_1 が満たされている．
- $\mathrm{G}^{\leqq\gamma}\phi$ ： ある状態から γ ステップまでの状態全てで ϕ が満たされている．
- $\mathrm{F}^{\leqq\gamma}\phi$ ： ある状態から γ ステップまでのどれかの状態で ϕ が満たされている．

PRISM ではある式の起こる確率を求めたり，確率付きで書かれた仕様を検証することができる．$\mathrm{P}_{\beta p}\ [\psi]$ は ψ が成り立つ確率が βp であることを表している．例えば,「ある状態 s から 10 ステップ経過するまで ϕ が成り立ち続ける確率は 25 %以下である」という仕様は $\mathrm{P}_{\leqq 0.25}\ [\mathrm{G}^{\leqq 10}\phi]$ となる．ここでのある状態 s とは，特に指定がない限り PRISM では全ての状態とみなされる．つまり，検証式の意味は「全ての状態において 10 ステップ経過するまで ϕ が成り立ち続ける確率は 25 %以下である」となる．

全ての状態からではなく初期状態から検証を開始したい場合には“ init”=> $\mathrm{P}_{\leqq 0.25}$ と記述し，ϕ が成り立つ状態から検証を始めたい場合には ϕ=> $\mathrm{P}_{\leqq 0.25}$ と記述する必要がある．

また，PRISM では検証だけではなく仕様を満たす確率も求めることが可能である．その場合には P=?[ψ] と記述すれば，ψ が成り立つ確率を計算することができる．

4. 送信予約システムのモデル化

既に行った研究 [3] では予約スロットの選択方法に重点を置き，実際の AIS での予約スロットの選択方法にできるだけ即した方法 1 つとそれを改良した 4 つのモデルを作成した．

実際の AIS における予約スロットの選択方法では基本的に選択するスロットは前のフレームで予約していたスロットと同じスロット箇所を選択する．しかし，そのままでは既に予約が重なっていた場合に常に重なり続けることになるため，6 回に 1 回は予約していたスロットの隣のスロットを選択するという方法となっている．

改良した 4 つのモデルの内の 1 つである det モデルでは次のように予約スロットを選択するようにした．

まず船舶は現在のフレームで予約していたスロット箇所の左右のスロットで既に予約されている数を左右それぞれ調べ，左右のスロットに予約されている数が同じ場合には，左右のスロットと現在のフレームで予約しているスロットの中からランダムにスロットを選択する．また，左右のスロットに予約されている数が違う場合，少ない方を 100%の確率で選択し，他の 2 箇所は絶対に選ばないようにした (図 1)．

作成した 5 つのモデルに対して予約が重ならないという意味での安全性の検証を行ったところ，実際の予約方法よりも改良版の方が良い結果となった．

5. Probabilistic Symmetry Reduction

[3] で行ったモデル検査ではすぐに状態爆発を起こしてしまい，極めて小さな規模についての検証しか行えなかった．本研究で

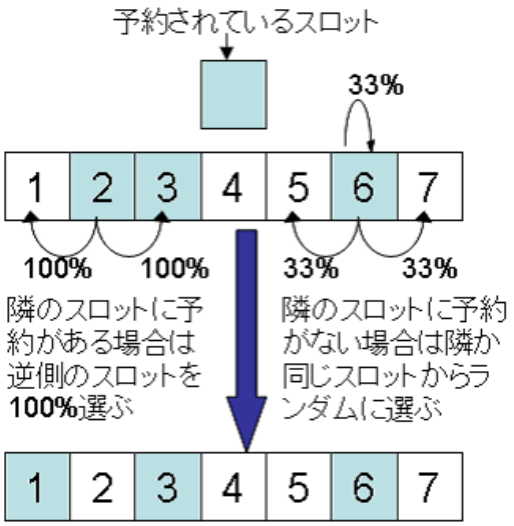

図 1　det モデルの説明
Fig. 1　det model

は det モデルに対して Probabilistic Symmetry Reduction [15] を施し，状態数削減を試みる．

5.1 基本概念

状態遷移系 M は，状態の集合 S と，遷移関係 $R \in S \times S$ の 2 項組 (S, R) として定義される．集合 S に対する写像 $\pi : S \to S$ が全単射であるとき，π を置換という．$(s, \bar{s}) \in R$ ならば $(\pi(s), \pi(\bar{s})) \in R$ のとき，π は R を保持すると言う．またこのとき，π を自己同型という．関数合成のもとで自己同型な群 G が与えられた時，s を $\bar{s}$ に写像するような G 上の置換があれば $(s, \bar{s}) \in \theta$ であるような S 上の同値関係が存在する．各同値類の代表元 1 つずつを含んだ状態集合を S' とする．各状態 $s \in S$ に対してそれの属する同値類の代表元を返す関数 $rep : S \to S'$ を定義することで，新たな遷移関係 $R' = \{(rep(s), rep(\bar{s})) \mid (s, \bar{s}) \in R\}$ を定義する．このようにして得られる状態遷移系 $M' = (S', R')$ を Quotient Model と呼ぶ．G 上の全ての置換が遷移関係 R を保持していることから，Quotient Model M' は元の状態遷移系 M と双模倣的である [16]．

5.2 AIS における Reduction

まず，遷移確率を無視して状態間の同値関係を定める．フレームのスロット数を m とし，フレームの左端と右端のスロットは繋がっていて全体がリング状になっているとみなす．4 節で構築したモデルにおける状態とは，あるフレームにおいてどのスロットが何隻に予約されているかを表したものとし，$s = \{r_1, \cdots, r_n\}$ と記述する．ただし，各 i $(1 \leqq i \leqq n)$ に対して $r_i = (t_i, v_i)$ とし，t_i は予約のあるスロット番号，v_i は t_i に予約している船舶の数とする．また，各 i $(1 \leqq i \leqq n)$ に対して $v_i = 1$ である場合に限り，$r_i = t_i$ と簡略化し記述する．

まず，船舶の識別子によらずフレーム内に予約されている箇所が全て一致した場合に同値とみなす．つまり，状態 s と s' における予約がそれぞれ $s = \{t_1, t_2, \cdots, t_h\}$，$s' = \{t'_1, t'_2, \cdots, t'_k\}$ の時に，$h = k$ かつ $t_l = t'_l$ $(1 \leqq l \leqq h)$ であれば同値であると

みなす.

次に「回転」と「反転」という操作を行う．状態 s に対してその操作を行った後の状態を s' とする．$s = \{r_1, \cdots, r_n\}, r_i = (t_i, v_i)$，$s' = \{r'_1, \cdots, r'_n\}, r'_i = (t'_i, v'_i)$ $(1 \leqq i \leqq n)$ とする．

例えば，図 2 の左図のように状態 $s_1 = \{1, 2, 5\}$，$s_2 = \{4, 5, 8\}$ である時に，状態 s_2 のフレーム全体を左へ 3 ずつずらすことで，$s_1 = s_2 = \{1, 2, 5\}$ となる．したがって，s_1 と s_2 は同値とである．

回転　フレーム全体を横にずらし，全ての予約スロットの位置をずらす操作を「回転」と呼ぶ.

状態の集合を S とする．α $(1 \leqq \alpha \leqq m)$ ずらす回転を表す S から S の関数 $shift_\alpha$ は以下のように定義される．

$$\begin{cases} t'_i = (t_i - \alpha + m) \bmod m \\ v'_i = v_{i-\alpha+m} \end{cases}$$

反転　フレームに対して左右対称移動を行う操作を「反転」と呼ぶ

反転を行う S から S の関数 $reverse$ は以下のように定義される．

$$\begin{cases} t'_i = (m - t_i + 1) \\ v'_i = v_{m-i+1} \end{cases}$$

(S, R) を AIS のもとのモデルの状態遷移系とするとき，$shift_\alpha$,$reverse$ は共に S 上における置換であり R を保持する．また，$shift_\alpha$，$reverse$ は関数合成に関して閉じている．したがって，$shift_\alpha$ と $reverse$ から合成されるすべての関数の集合を F とすれば $[s] = \{f(s) \mid f \in F\}$ が s の同値類の集合になる．任意の $f \in F$ は S 上の置換であり，R を保持する．$[s]$ の要素に対し以下のように順序付けを行う.

（1）　$t_1 < t'_1$ ならば $s \prec s'$

（2）　$t_1 = t'_1, \cdots, t_i = t'_i, t_{i+1} < t'_{i+1}$ となる i $(1 \leqq i < n)$ が存在すれば，$s \prec s'$

なお，$\forall i$ $(1 \leqq i \leqq n)$ に対して，$t_i = t'_i$ のときには $v_i = v'_i$ となるので，v_i については考慮する必要はない.

$s' \prec s$ となる $s' \in [s]$ が存在しないとき，この s を $[s]$ の代表元とし $rep(s)$ と記述する．このとき，$rep(s)$ は一意的に定まる.

5.3　遷移と確率

次に遷移確率まで考慮して Quotient Model を生成する．Reduction 前の状態遷移系を $M = (S, R, P)$，Reduction 後の状態遷移系を $M' = (S', R', P')$ とする．S', R' については 5.1 節で述べたように決めることができる.

P は遷移確率で $P : S \times S \to [0, 1]$ で定義される．また，全ての状態 $s \in S$ に対して $\Sigma_{\overline{s} \in S} P(s, \overline{s}) = 1$ (但し $(s, \overline{s}) \in R$) が満たされるとする.

det モデルでは全ての状態 $s, \overline{s} \in S$ および $f \in F$ に対して $P(f(s), f(\overline{s})) = P(s, \overline{s})$ が成り立つ．f は置換であり，ある状態から別の状態への置換前の遷移確率と置換後の対応する状態間の遷移確率を保持する．これは，次の状態を決める方法が現在の選択しているスロット及びその両隣のスロットのみに依存しており，また，この方法がどのスロットに対しても同じであるためである．したがって，Quotient Model における状態から状態への遷移確率は次のようになる.

全ての $s', \overline{s'} \in S'$ に対して

$$P'(s', \overline{s'}) = P(rep(s), rep(\overline{s})) \ (s' = rep(s), \overline{s'} = rep(\overline{s}))$$

6.　Quotient Model に対する検証と考察

6.1　設　　定

作成した Quotient Model に対して PRISM を用いて検証を行う．船舶は 3〜6 隻とし，1 フレームのスロット数は全ての場合で船舶数とスロット数の比率を同じするために，4 隻では 13 スロット，5 隻では 16 スロット，6 隻では 20 スロットとした．検証として用いた仕様は以下の 3 つである．仕様の中の $N_i frame$ $(i = 1, 2, 3)$ はフレーム数を表しており，検証時には 3 フレームと 20 フレームの 2 つを使用する．その理由としては AIS では，船が移動している場合には 3 回に 1 回，停泊している場合には 20 回に 1 回は他の船舶と予約スロットが重ならずに必ず送信が届く時間があればよいとされているためである．X_i $(i = 1, 2, 3)$ は確率を表し，検証段階でここに任意の値を代入して実行する.

Spec1　$\mathrm{P}_{\geqq X_1}[\mathrm{F}^{\leqq N_1 frame}$“ bst”$]$

N_1 フレーム中に 1 回以上は他の船舶と予約が重ならずに，情報の送信が正しく行える確率は X_1 以上である

Spec2　“ bst” $=>\mathrm{P}_{\leqq X_2}[\mathrm{G}^{\leqq N_2 frame}$“ bst”$]$

予約の重なりが起こってから N_2 フレーム連続で，予約スロットの衝突が起こる確率は X_2 以下である

Spec3　!“ bst” $=>\mathrm{P}_{\leqq X_3}[\mathrm{F}^{\leqq N_3 frame}$“ bst”$]$

予約スロットの重なりがない状態から N_3 フレーム中に 1 回以上は他の船舶と予約スロットが重なってしまう確率は X_3 以下である

6.2　検 証 結 果

Spec1，Spec2，Spec3 に対し確率 X を変動させ検証を行ったところ，結果は表 1 のようになった．また，比較として Reduction を行う前のモデルに対する検証結果を表 2 にまとめた．N はフレーム数を表し，表内の値は検証結果真となる確率の範囲を表している.

表 1　Reduction 後の検証結果

Table 1　Results of verification for the Quotient Model

船数	Spec1		Spec2		Spec3	
	N=3	N=20	N=3	N=20	N=3	N=20
3	0〜0.78	0〜0.999	0.25〜1	0.001〜1	0.42〜1	0.76〜1
4	0〜0.34	0〜0.999	0.69〜1	0.001〜1	1	1
5	0〜0.11	0〜0.999	0.89〜1	0.001〜1	1	1
6	0〜0.001	0〜0.99	0.999〜1	0.01〜1	1	1

表 2　Reduction 前の検証結果

Table 2　Results of verification for the original model

船数	Spec1		Spec2		Spec3	
	N=3	N=20	N=3	N=20	N=3	N=20
3	0〜0.78	0〜0.999	0.25〜1	0.001〜1	0.42〜1	0.76〜1

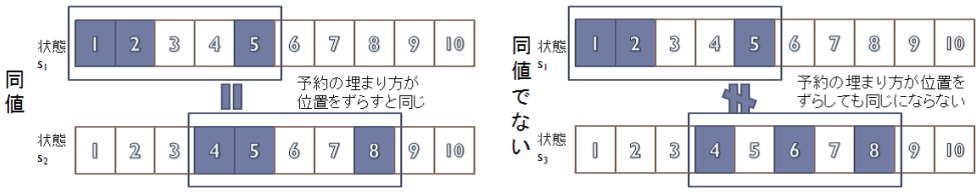

図 2　同値関係の例

Fig. 2　Equivalence relation

船舶が 3 隻では $N = 3$ にした場合「3 回に 1 回は正しく送信ができる確率」が 78%以上,「3 連続で予約が重なる確率」が 25%以下となり，安全であるとは言いがたい確率となった．しかし，$N = 20$ とした場合にはそれぞれの確率が 99%以上と 1%以下となり安全であるという結果であった．

また，表 1 と表 2 を比較すると，船舶が 3 隻の場合は，Reduction の前後で送信予約システムの双模倣性が保たれていることがわかる．

Reduction 前の 4 隻以上の検証は状態爆発により行えなかったが，船舶とスロットの比率が同じならば，船舶とスロットの数を増やしていっても，検証で得られる結果に多少の増減はあるとしても大きな差は出ないと予測を立てていた．

しかし，実験の結果，表 1 のフレーム数が 3 の場合を見ると，Spec1 では確率が徐々に減少していき,「3 回に 1 回は送信が重ならない確率」が悪くなっている．Spec2 に関しても「3 連続で送信が重なる確率」が船舶が増えるにつれ上昇しており，Spec3 に至っては「予約の重なりのない状態から予約が重なる確率」が 1 になり，予約は必ず重なるという結果になった．

一方，フレーム数が 20 の場合には 6 隻の時に確率が少し悪化するものの，Spec1 では 99%以上で Spec2 では 1%以下となり，安全といえる確率になった．

船舶の数を増加させると確率の悪化傾向が見られるのは，3.2 節で述べたように PRISM では検証時には初期値に含まれる状態全てからの仕様を満たす確率を調べていることが原因として挙げられる．

初期状態の遷移先に予約の重なりのある状態があり，かつ，その遷移確率が高ければ予約の重なる確率が高くなる．船舶の数が増えるとそのような初期状態が多数存在するため，検証結果として確率の悪化が見られることになった．

例えば Spec1 の検証において，全ての船舶が，あるスロットに予約している場合があるとする．船舶が 3 隻の場合では初期状態に依存することなく，次のフレームで 1 隻が同じスロット，あとの 2 隻が左右のスロットと別々のスロットに予約することが可能である．もし再び同じスロットを選択してしまったとしても，その次に別の予約を選ぶ確率は高い．そのため，3 回に 1 回は別々に予約できる確率は高くなる．

しかし，4 隻以上では全ての船舶があるスロットに予約をしている場合，次のフレームで全ての船舶が別々のスロットに予約することは不可能であり，その後も全ての船舶が別々のスロットに予約する確率は低い．そのため，3 回に 1 回は別々のスロットに予約する確率は低くなる．

6.3　解析結果

PRISM には初期状態による確率の違いをグラフで表示するような解析機能がある．その解析機能を利用して，船舶が 5 隻の場合に対してさらに詳しい解析を行った．その結果，次のような初期状態の場合に確率の低下が見られることがわかった．

- 半数以上の船舶が同じ箇所に予約している (図 3 の例 1)
- 初期状態から遷移した状態がいずれも予約の重なりがある (図 3 の例 2)
- 全ての船舶について、それが予約しているスロットの隣には別の船舶が予約している (図 3 の例 3)

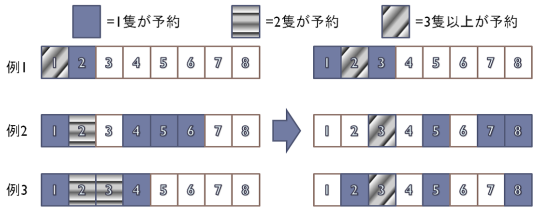

図 3　確率が低くなる例

Fig. 3　Examples of states with low probability

これらの場合を初期状態から除外し，Spec1 に対して再度検証を行うと，Reduction 前は 0.11 であった確率が Reduction 後には 0.5 程度まで上昇させることができた．したがって，船舶数を増やす場合，安全性を保つには初期状態の取り方を考える必要があることがわかった．

また，表 1 における Spec3 の船舶が 4 隻以上の場合に検証結果が 1 になっていた理由は上記の 2 番目の場合が当てはまる．つまり，初期状態としてそのような場合が少なくとも 1 つは含まれているために，船舶が 4 隻以上の場合の確率が 1 となった．

6.4　Reduction の効果

Symmetry Reduction を行う前のモデルと Reduction により得られた Quotient Model における状態数，遷移数，検証にかかる時間の変化を示す．

表 3　Reduction 後

Table 3　Results after reduction

船舶数/スロット数	状態数	総遷移数	検証時間 (sec)
3/10	14	87	0.0
4/13	92	1578	0.02
5/16	632	28997	0.672
6/20	5295	650531	15.094

船舶が 6 隻の場合には状態数が 4 桁分削減されており，Symmetry Reduction の効果が非常に高かったことを示している．

表 4　Reduction 前
Table 4　Results before reduction

船舶数/スロット数	状態数	総遷移数	検証時間 (sec)
3/10	1000	15360	0.421
4/13	28561	1011569	0.843
5/16	1048576	$85 * 10^6$	-
6/20	$64 * 10^6$	$15 * 10^9$	-

なお，PRISM は JAVA を介して実行されており，モデルの構築には JAVA が使用するメモリサイズをある程度確保する必要がある．本研究ではメモリサイズを大きくして 6 隻までの動作確認は行えたが，それ以上は計算機の性能的な問題から行えなかった．本研究で使用したのは CPU は AMD athlon(tm) 64 processor で周波数は 2.20GHz，メモリは 1.00GB である．

6.5 評　　価

Symmetry Reduction により状態数の大幅な削減ができ，船舶の数を増やした検証が可能になった．検証結果としては船舶の数が多くなると，初期状態によって予約の重なる確率が高くなるという結果となった．

一般的なモデル検査では状態爆発を避けるために，システムをそのままモデル化し検証を行うことはなく，規模の小さなモデルを作成し反例の有無を調べることで，実際のシステムの安全性などを検証する．この手法は確率を含まないシステムでは一般に成功している．

しかし，本研究のような確率的要素を含むシステムでは，量的な解析や検証が必要となり，これらはモデルの規模によって結果が変わる可能性があることが分かった．一方，確率を含むシステムは一般にモデル検査における計算量も多くなるため，規模的にかなり小さなものしか扱えない．したがって，効果的な Reduction の方法を考えていく必要がある．

7. おわりに

本研究では AIS で用いられている送信予約システムについて，実際の予約スロットの選択方法を改良したモデルに対して Probabilistic Symmetry Reduction を施すことで状態数を削減したモデルを作成し，検証を行った．

AIS のモデルは状態遷移確率も含め多くの対称性を持つ．そのため，Reduction 方法は回転と反転という簡単なものであったが，Reduction 前と比べると状態数は最大で 4 桁分の削減に成功し，船舶の数は倍に増やすことができた．

実験の結果，短い時間で予約スロットの重ならない確率や，連続でバースト状態が続く確率は船舶数が増えると悪くなっていくことがわかった．これは PRISM があらゆる状態を初期状態として検証を行うためと思われる．

今後は，解析結果で得られた確率の悪化する初期状態には別の予約方法を適用する等、予約スロットが連続で重なる確率が低くなるような工夫を行い，どんな初期状態においても高い安全性が得られるような方法を提案し，定量的な解析，検証を行いたい．

また，Probabilistic Symmetry Reduction による状態数削減を行っても，船舶が 6 隻の場合が限界であった．今後はより削減の行える手法の提案と適用を行い，より実際の船舶の数に近いモデルでの解析と検証を行っていきたい．

文　　献

[1] 塩地誠, 水城南海男, 矢内崇雄, 中島敏和, 小林健, 大塚賢, ”AIS 情報による海上交通管理システム高度化，沖電気工業 管制システム部，” 電子航法研究所第 3 回研究発表会，東京，2003.

[2] D. Parke, G. Norman and M. Kwiatkowska, ”PRISM: probabilistic model checking for performance and reliability analysis,” ACM SIGMETRICS Performance Evaluation Review, vol.36, pp.40-45, 2009.

[3] T.Toyoshima and K.Takahashi, ”Probabilistic model checking of an automatic identification system,” The 13th IASTED International Conference on Software Engineering and Applications, pp.45-52, 2009.

[4] T.Junttila, ”On the symmetry reduction method for petri nets and similar formalisms,” PhD thesis, Laboratory for Theoretical Computer Science, Helsinki University of Technology, 2003.

[5] J.Rintanen, ”Symmetry reduction for SAT representations of transition systems,” International Conference on Automated Planning and Scheduling/Artificial Intelligence Planning Systems, 2003.

[6] A.Emerson and P.Sistla, Symmetry and model checking, Formal Methods in System Design, Springer-verlag, Berlin, 1996.

[7] K. L. McMillan, ”Verification of infinite state systems by compositional model checking,” Proc. 10th IFIP WG 10.5 Advanced Research Working Conference on Correct Hardware Design and Verification Methods, Lecture Notes in Computer Science, vol.1703, pp.219-234, 1999.

[8] T.Sekizawa, T.Tsuchiya, K.Takahashi and T.Kikuno, ”Probabilistic model checking of the one dimensional ising model,” IEICE TRANS INF SYST, vol.E92-D, pp.1002-1011, 2009.

[9] D.D.E.Long, K.Gopinath and J.Elerath, ”Reliability modelling of disk subsystems with probabilistic model checking,” Storage Systems Research Center Technical Report UCSC-SSRC-09-05, 2009.

[10] A. Finkel, A. Petit, B. Berard, F. Laroussinie, L. Petrucci, M. Bidoit, P. McKenzie and P. Schnoebelen, Systems and software verification: model checking techniques and tools, Springer-Verlag, Berlin, 2001.

[11] F.Magniette, R.Lassaigne, T.Herault and S.Peyronnet, ”Approximate probabilistic model checking,” Proc. 5th International Conference on Verification, Model Checking and Abstract Interpretation, Lecture Notes in Computer Science, vol.2937, pp.73-84, 2004.

[12] H.Hermanns, J.Katoen, J.M.Kayser and M.Siegle, ”ETMCC: model checking performability properties of markov chains,” Proc. International Conference on Dependable Systems and Networks, p.673, 2003.

[13] H.L.S.Younes, ”Ymer: a statistical model checker,” Proc. 17th International Conference on Computer Aided Verification, Lecture Notes in Computer Science, Vol.3576, pp.429-433, Springer-Verlag, 2005.

[14] A.Bianco and L.de Alfaro, ”Model checking of probabilistic and nondeterministic systems,” Proc. 5th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, vol.1026, pp.499-513, Springer-Verlag, Berlin, 1995.

[15] D.Parker, G.Norman and M.Kwiatkowska, ”Symmetry reduction for probabilistic model checking,” Proc. 18th International Conference on Computer Aided Verification, Lecture Notes in Computer Science, vol.4144, pp.234-248, Springer-Verlag, 2006.

[16] A.Skou and K.Larsen, ”Bisimulation through probabilistic testing,” Information and Computation, vol.94, pp.1-28, 1991.